

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 3.41.1

1.3. Descripción:

Ghost es una herramienta de código abierto, para publicaciones profesionales creada en un stack moderno de Node.js. Diseñado para equipos que necesitan poder, flexibilidad y rendimiento.

1.4. Funcionalidades Core:

- **Crear post:** esta funcionalidad permite crear publicaciones que pueden ser expuestas en la vista usuario
- **Editar post:** esta funcionalidad permite editar las publicaciones creadas.
- **Eliminar post:** esta funcionalidad permite eliminar las publicaciones creadas.
- **Crear página:** esta funcionalidad permite crear páginas que pueden ser visibles en la vista usuario.
- **Editar página:** esta funcionalidad permite editar la página creada y cambiar su estado a no publicada.
- **Eliminar página:** esta funcionalidad permite eliminar la página creada.
- **Crear etiqueta:** esta funcionalidad permite crear etiquetas que pueden ser vinculadas a páginas y publicaciones
- **Editar etiqueta:** esta funcionalidad permite editar una etiqueta.
- **Eliminar etiqueta:** esta funcionalidad permite eliminar una etiqueta creada.
- **Invitar usuario:** esta funcionalidad permite agregar un usuario invitándolo mediante su email, y estableciendo un rol específico.
- **Editar usuario:** esta funcionalidad permite modificar la información existente del usuario, como nombre, redes sociales, etc.
- **Suspender usuario:** esta funcionalidad permite deshabilitar un usuario sin la necesidad de eliminarlo.
- **Activar usuario:** esta funcionalidad permite activar un usuario suspendido.
- **Eliminar usuario:** esta funcionalidad permite eliminar un usuario.

1.5. Diagrama de Arquitectura:

Link para ver diagrama de componentes:

[diagrama-de-componentes.png](#)

1.6. Diagrama de Contexto:

Link para ver diagrama de contexto:

[diagrama-de-contexto.png](#)

1.7. Modelo de Datos:

Link para ver modelo de datos:

[diagrama-de-dominio.png](#)

1.8. Modelo de GUI:

Link para ver modelo GUI

https://www.figma.com/file/aAZKh6nHZOCb7c83SVgZE1/ghost_3.41.1?node-id=0%3A1

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

- Implementar una distribución balanceada de categorías de pruebas en el sistema que permitan el crecimiento sostenible de la aplicación, disminuir tiempos de ejecución y costos de desarrollo. Esto incluye:
 - Pruebas exploratorias manuales
 - Pruebas de Extremo a Extremo
 - Pruebas con escenarios de validación de datos
 - Pruebas de regresión visual
 - Pruebas de reconocimiento
- Diseñar y actualizar inventarios de pruebas para las funcionalidades críticas de la aplicación mediante pruebas exploratorias en los módulos definidos en la sección 1:
 - Gestión de usuario de Staff.
 - Gestión de publicación o post
 - Gestión de página
- Ejecutar pruebas a la mayor cantidad posible de funcionalidades de la aplicación para minimizar la cantidad de fallos que pueden llegar a un ambiente productivo.
- Asegurar la calidad de la aplicación mediante la ejecución de pruebas automatizadas y manuales.

2.2. Duración de la iteración de pruebas:

La iteración tendrá duración de 8 semanas calendario, que iniciaran lunes de Semana 1 y terminan en Viernes de la Semana 8. Se tendrán 4 testers automatizadores de pruebas Senior, trabajando 8 horas semanales cada uno para un total de 64 horas/hombre. Los desarrolladores trabajaran *en un solo día las 8 horas*, realizando las tareas definidas en la Sección 2.5 (más abajo).

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

El recurso humano que estará trabajando en esta estrategia de pruebas se encuentra distribuido de la siguiente forma:

- 2 Ingenieros automatizadores Senior: Conocimiento fuerte en herramientas como Kraken y conocimiento medio en Playwright, Cypress y ResembleJS, y conocimientos fundamentales de servicios AWS.
- 2 Ingenieros automatizadores Senior: Conocimiento fuerte en herramientas como Playwright y conocimiento medio en Kraken, Cypress y ResembleJS.

Estos recursos tiene experiencia con aplicaciones web con bugs y defectos de interactividad y exposición de errores no controlados, lo que les da cierta versatilidad para entender aplicaciones web y montar soluciones de automatización en las mismas.

2.3.2. Recursos Computacionales

Los recursos computacionales que se usaran para la ejecución de la estrategia de pruebas se detallan en la siguiente tabla:

| Recurso | Cantidad | Valor Unitario | Total Horas | Total | Fuente |
|---|----------|--------------------|-------------|-----------|---------|
| Instancia EC2 Linux - t4g.xlarge in the US East (Ohio) - 8CPUs - 32 GB RAM | 1 | \$0,3 USD por hora | 720 | \$218 USD | Ver [1] |
| Computadoras empresariales Win10 - Core i7 - SSD 1T - RAM 16GB, portátiles. | 2 | N/A | N/A | N/A | N/A |
| Computadoras Macbook Pro - Core i7 - SSD 512GB - RAM 16GB. | 2 | N/A | N/A | N/A | N/A |

[1]: <https://calculator.aws/#/estimate?id=05875ed3f9d1b53d82f33c303606d2268814b4b7>

2.3.3. Recursos Económicos para la contratación de servicios/personal:

Para la ejecución de pruebas automatizadas se agregará recursos por \$218USD los cuales serán utilizados para contratar servicios de cómputo en la nube, puntualmente AWS. Ver sección 2.3.2.

2.4. TNT (Técnicas, Niveles y Tipos) de pruebas:

| Nivel | Tipo | Técnica | Objetivo |
|-------------|---|--------------------------------|--|
| Unitarias | Funcionales, positivas y negativas de caja blanca | API de automatización | Permitir el crecimiento sostenible del proyecto de software, probando unidades pequeñas de código. |
| Integración | Funcionales, positivas y negativas de caja blanca | API de automatización. | Asegurar la correcta integración de los componentes del software. Generar scripts que permitan trabajar de manera dinámica data para validación de formularios. |
| Sistema | Funcionales; positivas; de caja negra | Pruebas manuales exploratorias | Definir el flujo del sistema y el inventario de pruebas |

| | | | |
|--|---|---|---|
| | Funcionales; positivas; de caja negra | Record and Replay | Construir y automatizar escenarios de funcionalidades core de la aplicación. |
| | Funcionales; positivas y negativas; de caja negra | API de automatización (Kraken y Playwright) | Automatizar los flujos realizados en las pruebas manuales para que puedan ejecutarse de manera constante, automatizando pruebas tipo End to End. Automatizar procesos de pruebas de regresión visual entre hacia versiones futuras de Ghost. |
| | Funcionales; positivas y negativas; de caja negra | Pruebas de reconocimiento (Monkeys y Rippers) | Asegurar la calidad de la aplicación con flujos automatizados de pruebas, que estén corriendo en paralelo para detectar problemas en el software. Dar visibilidad de escenarios atípicos no provistos por los escenarios de pruebas preestablecidos |

2.5. Distribución de Esfuerzo

Tareas asignadas por semana:

Ingeniero Automatizador Senior 1:

Semana 1: Ejecución de pruebas exploratorias manuales a las funcionalidades asociadas a Posts. Inventario y verificación pruebas unitarias asociadas a código para Posts.

Semana 2: Configuración de la instancia de AWS con instancia de Ghost. Generación con Cypress de Ripper+Monkey especializados en secciones de Posts. Instalación en instancia AWS de scripts de reconocimiento hechos

Semana 3: Arranque generación scripts de pruebas Extremo a Extremo (E2E) para funcionalidades asociadas a Posts con Playwright y Kraken.

Semana 4: Cierre scripts de pruebas Extremo a Extremo (E2E) para funcionalidades asociadas a Posts con Playwright y Kraken. Code review scripts E2E hechos por los demás Ingenieros para aprobación.

Semana 5: Arranque generación scripts de pruebas con escenarios de validación de datos de formularios asociados a Posts con Playwright y Kraken.

Semana 6: Cierre scripts de pruebas con escenarios de validación de datos de formularios asociados a Posts con Playwright y Kraken. Code review scripts tipo validación datos hechos por los demás Ingenieros para aprobación.

Semana 7: Generación (y cierre) scripts de pruebas de regresión visual (o VRT, basándose en scripts anteriores) para funcionalidades asociadas a Posts con Playwright y Kraken. Code review scripts VRT hechos por los demás Ingenieros para aprobación. Análisis preliminar de outputs de pruebas de reconocimiento en Instancia AWS.

Semana 8: Recopilación de información obtenida en el proceso de pruebas de funcionalidades Posts. Generación diapositivas para presentación ejecutiva. Redacción reportes PDF sobre funcionalidades Posts. Realización presentación ejecutiva.

Ingeniero Automatizador Senior 2:

Semana 1: Ejecución de pruebas exploratorias manuales a la funcionalidad asociadas a Páginas Ghost. Inventario y verificación pruebas unitarias asociadas a código para Páginas Ghost.

Semana 2: Apoyo a configuración de la instancia de AWS. Generación con Cypress de Ripper+Monkey especializados en secciones de Páginas Ghost. Instalación en instancia AWS de scripts de reconocimiento hechos.

Semana 3: Arranque generación scripts de pruebas Extremo a Extremo para funcionalidades asociadas a Páginas Ghost, con Playwright y Kraken.

Semana 4: Cierre scripts de pruebas Extremo a Extremo (E2E) para funcionalidades asociadas a Páginas Ghost con Playwright y Kraken. Code review scripts E2E hechos por los demás Ingenieros para aprobación.

Semana 5: Generación scripts de pruebas con escenarios de validación de datos de formularios asociados a Páginas Ghost con Playwright y Kraken.

Semana 6: Cierre scripts de pruebas con escenarios de validación de datos de formularios asociados a Páginas Ghost con Playwright y Kraken. Code review scripts tipo validación datos hechos por los demás Ingenieros para aprobación.

Semana 7: Generación (y cierre) scripts de pruebas de regresión visual (basándose en scripts anteriores) para funcionalidades asociadas a Páginas Ghost con Playwright y Kraken. Code review scripts VRT hechos por los demás Ingenieros para aprobación. Apoyo a análisis preliminar de outputs de pruebas de reconocimiento en Instancia AWS.

Semana 8: Recopilación de información obtenida en el proceso de pruebas de funcionalidades Páginas Ghost. Apoyo generación diapositivas para presentación ejecutiva. Redacción reportes PDF sobre funcionalidades Páginas. Realización presentación ejecutiva.

Ingeniero Automatizador Senior 3:

Semana 1: Ejecución de pruebas exploratorias manuales a las funcionalidades asociadas a Etiquetas. Inventario y verificación pruebas unitarias asociadas a código para Etiquetas.

Semana 2: Generación con Cypress de Ripper+Monkey especializados en secciones de Etiquetas. Instalación en instancia AWS de scripts de reconocimiento hechos.

Semana 3: Arranque generación scripts de pruebas Extremo a Extremo para funcionalidades asociadas a Etiquetas, con Playwright y Kraken.

Semana 4: Cierre scripts de pruebas Extremo a Extremo (E2E) para funcionalidades asociadas a Etiquetas con Playwright y Kraken. Code review scripts E2E hechos por los demás Ingenieros para aprobación.

Semana 5: Generación scripts de pruebas con escenarios de validación de datos de formularios asociados a Etiquetas con Playwright y Kraken.

Semana 6: Cierre scripts de pruebas con escenarios de validación de datos de formularios asociados a Etiquetas con Playwright y Kraken. Code review scripts tipo validación datos hechos por los demás Ingenieros para aprobación.

Semana 7: Generación (y cierre) scripts de pruebas de regresión visual (basándose en scripts anteriores) para funcionalidades asociadas a Etiquetas con Playwright y Kraken. Code review scripts VRT hechos por los demás Ingenieros para aprobación.

Semana 8: Recopilación de información obtenida en el proceso de pruebas de funcionalidades Etiquetas. Redacción reportes PDF sobre funcionalidades Etiquetas. Realización presentación ejecutiva.

Ingeniero Automatizador Senior 4:

Semana 1: Ejecución de pruebas exploratorias manuales a las funcionalidades asociadas a Perfiles de Usuarios (abreviado en adelante como Usuarios). Inventario y verificación pruebas unitarias asociadas a código para Usuarios.

Semana 2: Generación con Cypress de Ripper+Monkey especializados en secciones de Usuarios y página principal de administración Ghost. Instalación en instancia AWS de scripts de reconocimiento hechos.

Semana 3: Arranque generación scripts de pruebas Extremo a Extremo para funcionalidades asociadas a Usuarios, con Playwright y Kraken.

Semana 4: Cierre scripts de pruebas Extremo a Extremo (E2E) para funcionalidades asociadas a Usuarios con Playwright y Kraken, y code review scripts E2E hechos por los demás Ingenieros para aprobación.

Semana 5: Generación scripts de pruebas con escenarios de validación de datos de formularios asociados a Usuarios con Playwright y Kraken.

Semana 6: Cierre scripts de pruebas con escenarios de validación de datos de formularios asociados a Usuarios con Playwright y Kraken. Code review scripts tipo validación datos hechos por los demás Ingenieros para aprobación.

Semana 7: Generación (y cierre) scripts de pruebas de regresión visual (basándose en scripts anteriores) para funcionalidades asociadas a Usuarios con Playwright y Kraken. Code review scripts VRT hechos por los demás Ingenieros para aprobación.

Semana 8: Recopilación de información obtenida en el proceso de pruebas de funcionalidades Usuarios. Redacción reportes PDF sobre funcionalidades Usuarios. Realización presentación ejecutiva.

Detalle cronograma para estrategia de prueba:

| Semanas Calendario / Acciones | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------------------------------------|---|---|---|---|---|---|---|---|
| Pruebas exploratorias manuales | | | | | | | | |
| Configuración de instancia AWS | | | | | | | | |
| Scripts de Pruebas de reconocimiento | | | | | | | | |

| | | | | | | | | |
|---|--|--|--|--|--|--|--|--|
| Pruebas de extremo a extremo | | | | | | | | |
| Pruebas con escenarios de validación de datos | | | | | | | | |
| Pruebas de regresión visual | | | | | | | | |
| Análisis de resultados de pruebas de reconocimiento | | | | | | | | |
| Recolección de datos de todas las pruebas hechas | | | | | | | | |
| Generación de documentación para reportes | | | | | | | | |
| Presentación ejecutiva | | | | | | | | |
| Instancia 1 AWS | | | | | | | | |

A continuación, se relacionan las horas de los recursos utilizados para la estrategia de pruebas por semana:

| Semanas / Recurso | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------------------------|---|----|----|-----|-----|----|----|---|
| Tester automatizador Senior 1 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Tester automatizador Senior 2 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Tester automatizador Senior 3 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Tester automatizador Senior 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| Instancia_1 EC2 AWS | 0 | 84 | 84 | 168 | 168 | 84 | 84 | 0 |